

This listing of claims will replace all prior versions, and listings, of claims in the application.

LISTING OF CLAIMS:

1. (Currently Amended) A method for controlling the synchrony of interprocess communication (IPC) which uses:
 - a. the original sender of the communication, called the source;
 - b. the ultimate receiver of the communication, called the destination;
 - c. a kernel process that is capable of delivering communications and unblocking the source of the communication;
 - d. one or more processes, called monitors, to which the kernel process may choose to deliver a communication to, rather than the destination, and which have the ability to request control of when a source is unblocked, called the controlling monitor of the source[.], wherein the method for controlling the synchrony of interprocess communication comprise the following steps:

[d.] when the communication is sent by the source, the source is blocked by the kernel process;

[e.] the kernel either delivers the communication to the destination or to a monitor;

[f.] the monitor may request that the kernel process make it the controlling monitor of the source;

[g.] the monitor forwards the communication to the destination specifying the source of the communication, and the monitor is blocked by the kernel process until the communication is delivered either to the destination or another monitor;

[h.] when the communication is delivered to the destination, the kernel process either: (1) signals the controlling monitor of the specified source, if any, or (2) unblocks the specified source;

[i.] the controlling monitor of the source can request that the kernel process unblock the source at any time after being signalled that the communication was delivered to the destination[.];

whereby the interprocess communication appears to the destination and to the source to have the same semantics regardless of whether the communication is sent directly from the source to the destination without being examined by the monitor, or the communication is sent from the source to the destination via the monitor and the monitor examines the communication.

2. (Original) A method as claimed in claim 1, wherein the monitor may change the identity of the source that is seen by the destination while maintaining the identity of the original source for unblocking purposes.

3. (Original) A method as claimed in claim 1, wherein the monitor may change the identity of the destination to which the communication is to be delivered.

4. (Currently Amended) A method as claimed in claim 1, wherein the communication embodies a message, and said message embodied in the communication is not delivered to the monitor.

5. (Original) A method as claimed in claim 1, wherein the timeouts on the communication are interpreted as being relative to the behaviors of the source and destination.

6. (Currently Amended) A method as claimed in claim 5, wherein the kernel process does not deliver the communication to the destination or a monitor until the destination is ready to receive the communication and the destination timeout (~~i.e., the timeout set by the source on the amount of time it will wait for the destination to become ready to receive the communication~~) has not expired, said destination timeout being the timeout set by the source on the amount of time it will wait for the destination to become ready to receive the communications.

7. (Original) A method as claimed in claim 5, wherein the kernel process delivers the communication to the monitor as soon as the monitor is ready, and the monitor checks for the destination timeout's expiration.

8. (Currently Amended) A method as claimed in claim 5, wherein the kernel process verifies the source timeout (~~i.e., the timeout set by the destination on the amount of time it will wait for the source to initiate the communication~~) has not expired before sending the communication to the destination or a monitor, said source timeout being the timeout set by the destination on the amount of time it will wait for the source to initiate the communication.

9. (Original) A method as claimed in claim 1, wherein multiple monitor processes may claim to be the controlling monitor of a source.

10. (Original) A method as claimed in claim 1, wherein the kernel process may authorize a monitor's permission to be the controlling monitor of a particular source.

11. (Original) A method as claimed in claim 10, wherein the kernel process may authorize a monitor's permission to be the controlling monitor of any source.

12. (Original) A method as claimed in claim 1, wherein the controlling monitor for a particular source may be stored in the kernel.

13. (Original) A method as claimed in claim 9, wherein a sequence of controlling monitors for a particular source may be stored in the kernel.

14. (Original) A method as claimed in claim 1, wherein the identity of the controlling monitor of a particular source is passed in the IPC to the destination.

15. (Original) A method as claimed in claim 14, wherein the sequence of controlling monitors for a particular source is stored by the controlling monitor storing its controlling monitor predecessor for each source.

16. (Original) A method as claimed in claim 1, wherein the controlling monitor for a source is implemented by changing the original source to the controlling monitor and the controlling monitor stores the identity of the original source.

17. (Currently Amended) A method as claimed in claim 16, wherein a sequence of controlling monitors for a particular source is implemented as a sequence of original source changes in the monitors where the last in the sequence is the ~~true~~ original source.

18. (Currently Amended) A method as claimed in claim 1, wherein the monitors are implemented as threads ~~in the same process~~.

19. (Currently Amended) A method as claimed in claim 1, wherein the monitors are implemented as procedures ~~in the same process~~.

20. (Original) A method as claimed in claim 1, wherein the monitor procedures are in the kernel process.

21. (Currently Amended) A method of performing interprocess communications (IPCs), comprising the steps of:

an IPC process receiving IPC requests, each of the IPC requests including a source identifier identifying a source and a destination identifier identifying a destination;

building IPCs in response to the requests;

transmitting the IPCs from the sources to the destinations;

intercepting and examining selected ones of the IPCs; and

controlling the synchrony of the IPCs so that each IPC ~~appear~~ appears to the source and to the destination to ~~be implemented according to~~ have the same semantics regardless of whether the IPC is ~~intercepted and examined~~ sent directly from the source to the destination without being intercepted and examined by the monitor, or the communication is sent from the source to the destination via the monitor and the monitor examines the communication.

22. (Original) A method according to Claim 21, wherein:

the step of building and transmitting IPCs includes the step of using a kernel to build and transmit the IPCs;

the step of intercepting and examining selected ones of the IPCs includes the step of using monitors to intercept and examine the selected ones of the IPCs; and

the controlling step includes the step of using the monitors as extensions of the kernel so that the IPCs appear to the sources and to the destinations to be implemented according to the same semantics regardless of whether a monitor is used or not used to intercept and examine the IPCs.

23. (Original) A method according to claim 22, wherein, for each IPC, one or more of the monitors manages the identity of the source for the IPC.

24. (Currently Amended) A method according to Claim 22, further comprising the step of, for each IPC, blocking the source for the IPC at selected times, and wherein one or more of the monitors ~~manages~~ determines when the source is unblocked.

25. (Original) A method according to Claim 22, wherein, for each IPC, one of the monitors manages whether the monitor is notified when the IPC reaches the destination for the IPC.

26. (Original) An interprocess communications (IPCs) system, comprising:

a processor having an IPC process for receiving IPC requests, each of the IPC requests including a source identifier identifying a source for the IPC and a destination identifier identifying a destination for the IPC;

means for building IPCs in response to the requests;

means for transmitting the IPCs from the sources to the destinations;

means for intercepting and examining selected ones of the IPCs; and

a controller for controlling the synchrony of the IPCs so that each IPC ~~appear~~ appears to the source and to the destination for the IPC to ~~be implemented~~ according to have the same semantics regardless of whether the IPC is ~~intercepted and examined~~ sent directly from the source to the destination without being intercepted and examined by the monitor, or the communication is sent from the source to the destination via the monitor and the monitor examines the communication.

27. (Original) A system according to Claim 26, wherein:

the means for building and transmitting IPCs includes a kernel to build and transmit the IPCs;

the means for intercepting and examining selected ones of the IPCs include monitors to intercept and examine the selected ones of the IPCs; and

the monitors are used as extensions of the kernel so that the IPCs appear to the sources and to the destinations to be implemented according to the same semantics regardless of whether a monitor is used or not used to intercept and examine the IPCs.

28. (Original) A system according to claim 26, wherein, for each IPC, one or more of the monitors manages determines the identity of the source for the IPC.

29. (Currently Amended) A system according to Claim 26, wherein, for each IPC, the source for the IPC is blocked at selected times, and one or more of the monitors ~~manages~~ determines when the source is unblocked.

30. (Original) A system according to Claim 26, wherein, for each IPC, one of the monitors manages whether the monitor is notified when the IPC reaches the destination for the IPC.

31. (Currently Amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for performing interprocess communications (IPCs), said method steps comprising:

an IPC process receiving IPC requests, each of the IPC requests including a source identifier identifying a source and a destination identifier identifying a destination;

building IPCs in response to the requests;

transmitting the IPCs from the sources to the destinations;

intercepting and examining selected ones of the IPCs; and

controlling the synchrony of the IPCs so that each IPC ~~appear~~ appears to the source and to the destination to ~~be implemented~~ according to have the same semantics regardless of whether the IPC is ~~intercepted and examined~~ sent directly from the source to the destination without being intercepted and examined by the monitor,

or the communication is sent from the source to the destination via the monitor and the monitor examines the communication.

32. (Original) A program storage device according to Claim 31, wherein:

the step of building and transmitting IPCs includes the step of using a kernel to build and transmit the IPCs;

the step of intercepting and examining selected ones of the IPCs includes the step of using monitors to intercept and examine the selected ones of the IPCs; and

the controlling step includes the step of using the monitors as extensions of the kernel so that the IPCs appear to the sources and to the destinations to be implemented according to the same semantics regardless of whether a monitor is used or not used to intercept and examine the IPCs.

33. (Original) A program storage device according to claim 32, wherein, for each IPC, one or more of the monitors manages the identity of the source for the IPC.

34. (Currently Amended) A program storage device according to Claim 32, further comprising the step of, for each IPC, blocking the source for the IPC at selected times, and wherein one or more of the monitors ~~manages~~ determines when the source is unblocked.

35. (Original) A program storage device according to Claim 32, wherein, for each IPC, one of the monitors manages whether the monitor is notified when the IPC reaches the destination for the IPC.